Prepared for: MAS.834 TANGIBLE INTERFACE.
Prepared by: Frank Appiah,
September/December 2025.
Proposal Design: FoodieDesign

EXECUTIVE SUMMARY

Objective

This project is based on a restaurant setting where interaction takes place with a keyboard and a screen. This is used in the processing of food menu request. On a table is a keyboard or off is a wireless keyboard. Based on the tag description of each menu item, it is feasible to use attached keyboard to select a food menu description.

Goals

There is a need to stimulate appetite with eaters in queue for their turn to buy food or wait around the table not wondering what to eat. Turns on/off the table can be very convincing to waiters who are called every single time to attend to the next eater around the dining table.

Solution

Several images are used to represent many food menu and any time a key is pressed, it changes to stimulate a new food menu for the eaters selection. A table screen will be used to stimulate eaters appetite at front desk. Changing screens will alert waiters to attend to the screens continuously.

Project Outline

1. Uses several images to stimulate food menu.
2. Key presses to show change of menu item to select from.
3. Coding of the prototype is done with processing.
4. Table screen is a monitor that displays the selected menu item.

APPLICATION SCREENS

7
Three environments for continuously making eatables like food.
Renewable Energy.
4
>>>>
New agriculture seed growing methods are already heading the way.
7
Three environments for continuously making eatables like food.
Renewable Energy.

CODING DETAILS

Objective

```
iimport processing.video.*;
import java.util.Arrays;
import java.io.*;
import beads.*;
import java.net.URL;
import java.awt.*;
import processing.core.PApplet;
import java.util.Timer;
import java.util.TimerTask;
import java.awt.Rectangle;
import java.awt.Robot;
import java.awt.AWTException;
import java.awt.event.MouseEvent;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.JOptionPane;
import processing.pdf.*;

boolean titleOnly=true, actionTypescreen=false, titleScreen=false, startScreen=false, clipScreen=false,
fullScreen=false;
boolean storyScreen=false, rulesScreen=false, aboutScreen=false, menuScreen=false, settingScreen=false;
PImage titleonly;
String soundPath=null;
int timerPlays[]={20, 50, 60, 50, 100, 35, 55, 90};
Robot r;
PGraphicsPDF pdf=null;
String fontType = "data/sans.tff";

void setup() {
```

```
  fullScreen();
  surface.setLocation(width/2-800/2, height/2-600/2);
  surface.setSize(800, 600);
  surface.setResizable(true);
  smooth();
  ac = AudioContext.getDefaultContext();
  try {
    r = new Robot();
  }
  catch(AWTException ee) {
  }
  //textFont(createFont(fontType, 12));
  titleonly=loadImage("foodie0.png");
  // pdf = (PGraphicsPDF) createGraphics(width/3, height/3, PDF, "data/foodie.pdf");
  frameRate(30);
}

String pdfPath=null;
String loadPdf() {
  selectInput("Select an image to render or clip or animate", "fileSelected");
  return pdfPath;
}

int pages=11, page=0, pagination=5, ind=0;
PImage[] pagind=new PImage[5];
PImage pageImg;
void keyPressed() {
  if (key == CODED) {
    if (keyCode == ESC) {
      exit();
    }
  } else if (key == 'F' || key =='f' ) {
    if (pageImg!=null) {
      pageImg.filter(color((int)random(120, 200), (int)random(12, 200), (int)random(120, 255)));
    }
  } else if (key == 'P' || key =='p' ) {

  } else if (key == 'N' || key =='n' ) {

    if (page < pages) {
```

KEYS:

F Key

P Key

N Key

M Key

```
      pageImg=loadImage("foodie"+page+".jpg");

      ind=pagination-page;

      pagind[0]=loadImage("foodie"+(page)+".jpg");
      pagind[1]=loadImage("foodie"+(page+1)+".jpg");
      pagind[2]=loadImage("foodie"+(page+2)+".jpg");
      pagind[3]=loadImage("foodie"+(page+3)+".jpg");
      pagind[4]=loadImage("foodie"+(page+4)+".jpg");
    } else page=0;
    page++;
  } else if (key == 'm' || key =='M' ) {
    if (page <= pages && page>=0) {

      pageImg=loadImage("foodie"+page+".jpg");

      pagind[0]=loadImage("foodie"+(page)+".jpg");
      pagind[1]=loadImage("foodie"+(page+1)+".jpg");
      pagind[2]=loadImage("foodie"+(page+2)+".jpg");
      pagind[3]=loadImage("foodie"+(page+3)+".jpg");
      pagind[4]=loadImage("foodie"+(page+4)+".jpg");
    } else page=0;
    page--;
  }
}

void fileSelected(File pdfFile) {
  if (pdfFile.exists()) {
    try {
      pdfPath=pdfFile.getAbsolutePath();
    }
    catch(Exception e) {
    }
  }
}

Timer sndTimer, splashTimer, arrowTimer;
int arrowSpace=20, arrowSpace2=20;

void createArrowAnim() {
```

```
  arrowTimer=new Timer("SmaEats", true);
  arrowTimer.schedule(new TimerTask() {
    public void run() {
      textSize(150);
      if (arrowSpace < displayWidth-200) {
        pushStyle();
        fill((int) random(10, 255));
        text("> > > >", arrowSpace, displayHeight-150);

        popStyle();
        arrowSpace+=20;
      }
    }
  }
  , 0, 2*timerPlays[(int)random(0, 6)]);
}

boolean splashed=false;
void draw() {
  background(0);

  if (pageImg!=null)
  {
    if ((millis()%1)==0) {
      image(pageImg, width/2- (width-300)/2, height/2-(height-140)/2, (int)random(width-300, width-299),
(int)random(height-140, height-138) );
      if (pagind[0]!=null)
        image(pagind[0], 100, height-150, 100, 100);
      if (pagind[1]!=null)
        image(pagind[1], 250, height-120, 100, 100);
      if (pagind[2]!=null)
        image(pagind[2], 400, height-120, 100, 100);
      if (pagind[3]!=null)
        image(pagind[3], 550, height-120, 100, 100);
      if (pagind[4]!=null)
        image(pagind[4], 700, height-120, 100, 100);
      pushStyle();
      fill((int)random(10, 100), (int)random(100, 230), (int)random(10, 255));
      text(">", 100, height-130);
      text(""+(page), width-140, 130);
```

```
      popStyle();
    }
  } else
  {
    //splash
    image(titleonly, 0, 0, width, height);
  }

  if ((millis()%50)==0 && !splashed) {
    surface.setSize(displayWidth, displayHeight);
    int  p[]={width/2, height/2};
    int  p1[]={width, height};
    surface.placeWindow(p, p1);
    surface.setResizable(false);
    splashed=true;
  }
  if (!fullScreen) {
    fullScreen=true;
    createArrowAnim();
    // createSoundTimer();
  }
}

void createSplashTimer() {
  splashTimer=new Timer("SmaEatsSplash", true);
  splashTimer.schedule(new TimerTask() {
    public void run() {
    }
  }
  , 0, timerPlays[(int)random(0, 6)]);
}


void createSoundTimer() {
  sndTimer=new Timer("SmaEatsSnd", true);
  sndTimer.schedule(new TimerTask() {
    public void run() {
      if (soundPath!=null)
      playSoundPath(soundPath);
      else
```

```
        playSound("sound60.wav");
    }
  }
  , 0, 2*timerPlays[(int)random(0, 6)]);
}

void playSound(String sound) {

  try
  {
    //production
    String os=System.getProperty("os.name");
    String doc="";
    if (os.startsWith("Mac") || os.equalsIgnoreCase("Linux"))
    {
      doc= "/Users/"+System.getProperty("user.name")+"/Documents/Foodie/data/beatbox.mp3";
      File soundCheck=new File(doc);
      if (soundCheck.exists())
      {
        doc="/Users/"+System.getProperty("user.name")+"/Documents/Foodie/data";
      } else
      {
        doc= "/Applications/Foodie/data/beatbot.mp3";
        soundCheck=new File(doc);
        if (soundCheck.exists())
        {
          doc="/Users/Applications/Foodie/data/";
        } else
          doc=null;
      }
    }

    // print("CD directory:::"+doc);
    File str;

    if (doc==null) {
      str=new File(new String("/Users/frankappiah/Documents/Processing/Foodie")+"/data/"+sound);
    } else {
      str=new File(new String(doc+sound));
    }
```

```
    soundPlayer = new SamplePlayer(SampleManager.sample(str.getAbsolutePath()));
    soundPlayer.setToLoopStart();
    //soundPlayer.setLoopType(SamplePlayer.LOOP_ALTERNATING);
    Gain g = new Gain(2, 0.2);
    g.addInput(soundPlayer);
    ac.out.addInput(g);

    ac.start();
  }
  catch(Exception io) {
    print("Err:"+io.toString());
  }
}



SamplePlayer soundPlayer;
AudioContext ac;
void playSoundPath(String soundpath) {

  try
  {
    soundPlayer = new SamplePlayer(SampleManager.sample(soundpath));
    soundPlayer.setToLoopStart();
    //soundPlayer.setLoopType(SamplePlayer.LOOP_ALTERNATING);
    Gain g = new Gain(2, 0.2);
    g.addInput(soundPlayer);
    ac.out.addInput(g);

    ac.start();
  }
  catch(Exception io) {
    print("Err:"+io.toString());
  }
}
```